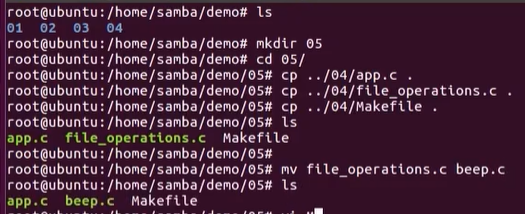
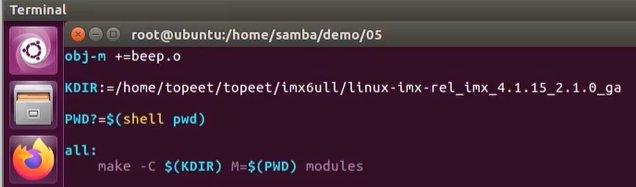


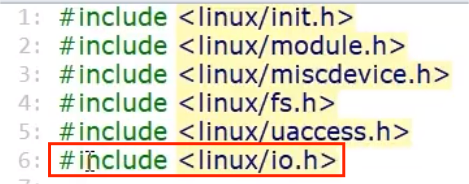
本节课将在上一节实践课的基础上完成：

将上一节课的三个文件拷贝过来，并且给驱动文件重命名为beep.c

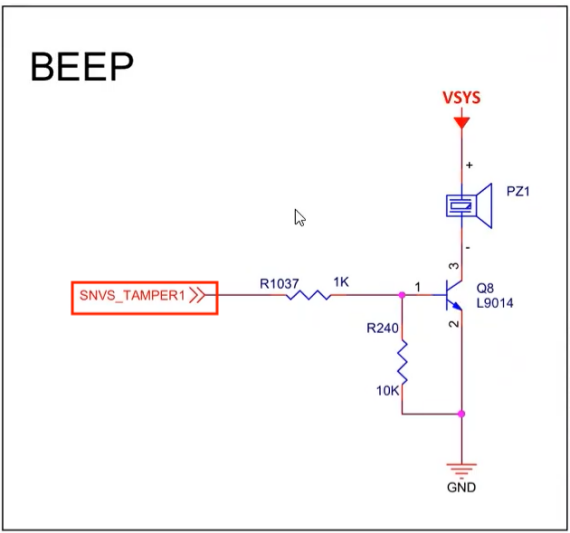


修改Makefile文件：



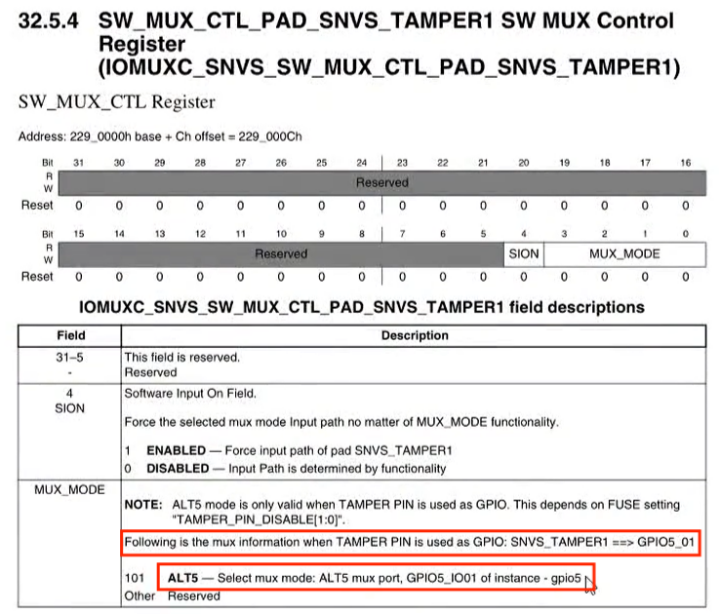
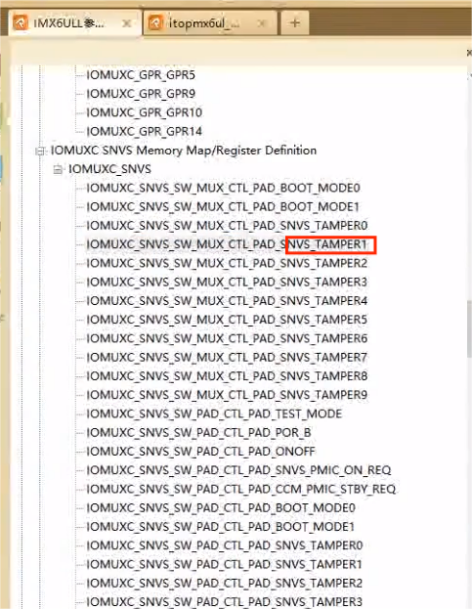
实现对蜂鸣器的控制：  
 添加头文件：  


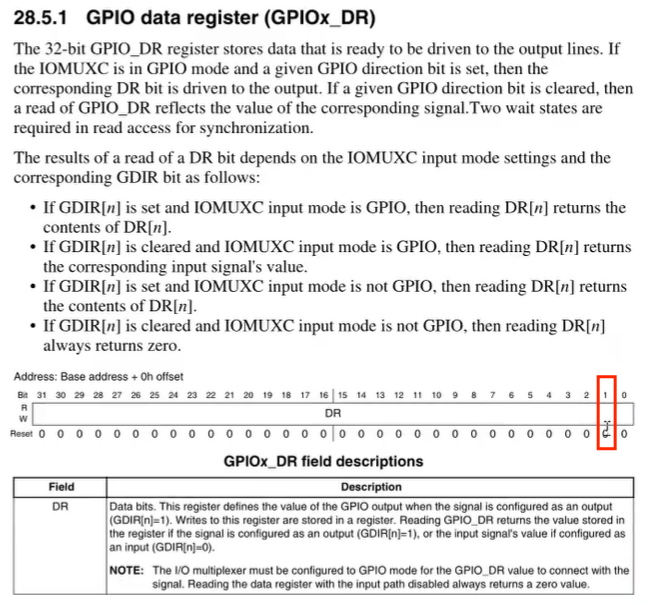
查阅手册，确定蜂鸣器对应的寄存器地址：



由原理图可知，其可操作管脚为：SNVS\_TAMPER1，然后打开手册：由于我们内核里面已经有了蜂鸣器的驱动了，所以像 复用关系的寄存器 就不用设置了，我们直接设置数据寄存器即可。

我们可以看到：TAMPER1这个管脚的复用是5：

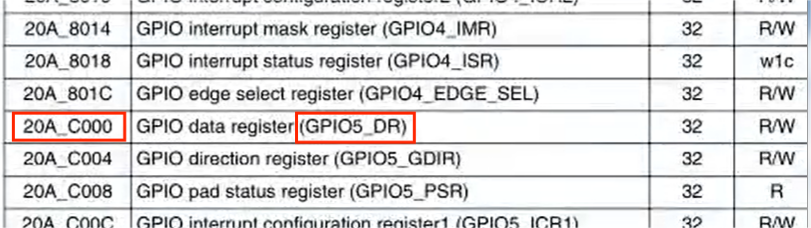




可见：操作引脚就是GPIOx\_01。

我们将数据手册打开，找到GPIO\_DR寄存器：

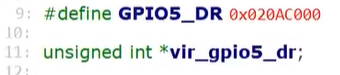
我们先来确定一些GPIO5\_01引脚的地址寄存器是多少：



因此，回到程序，继续编写代码：

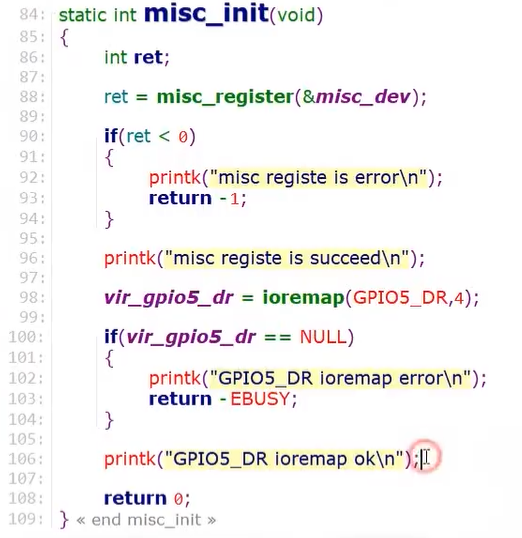
根据手册，定义GPIO\_DR 的 物理地址为0x020AC000

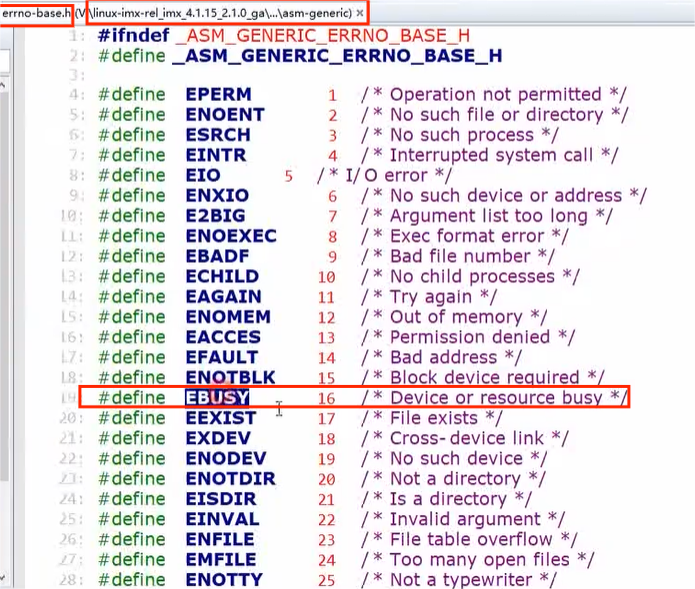
再定义一个无符号整型vir\_gpio5\_dr，，用于保存虚拟映射之后的虚拟地址。



我们在驱动加载的时候，就会执行misc\_init函数，我们就在该函数里面映射我们的虚拟地址即可：

这里 EBUSY含义：其代表16，表示设备或资源忙。

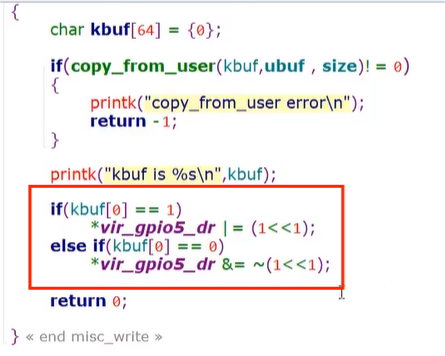


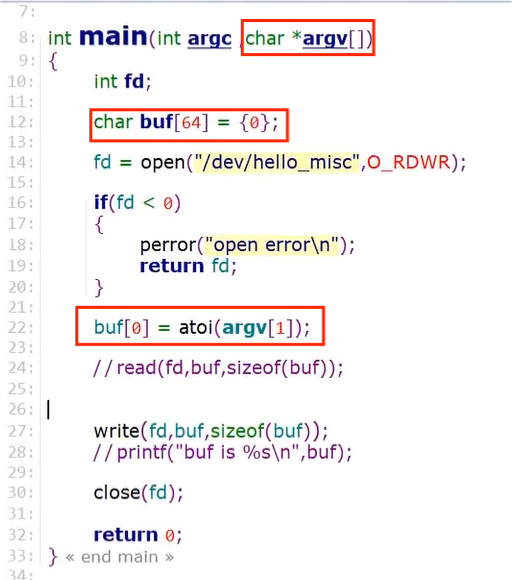


取消映射：  


到此，完成了物理映射的过程，下面进行蜂鸣器开关的逻辑程序：

应用层打开设备，然后向设备节点写1，所以需要修改驱动的write函数：



应用层编写：  


这里atoi函数用于将终端的argv参数转换成字符串。然后保存在buf里面。

编译拷贝到开发板：



在开发板加载驱动，执行应用层程序：

